

Real-Time Monitoring Method for Cow Rumination Behavior Based on Edge Computing and Improved MobileNet v3

ZHANG Yu¹, LI Xiangting¹, SUN Yalin², XUE Aidi^{1,3}, ZHANG Yi¹,
JIANG Hailong¹, SHEN Weizheng^{1*}

(1. College of Electrical and Information, Northeast Agricultural University, Harbin 150030, China; 2. Harbin Aerospace Stellar Data System Technology Co., Ltd, Harbin 150030, China; 3. Harbin Electric Machinery Company Co., Ltd., Harbin 150030, China)

Abstract: [Objective] Real-time monitoring of cow ruminant behavior is of paramount importance for promptly obtaining relevant information about cow health and predicting cow diseases. Currently, various strategies have been proposed for monitoring cow ruminant behavior, including video surveillance, sound recognition, and sensor monitoring methods. However, the application of edge device gives rise to the issue of inadequate real-time performance. To reduce the volume of data transmission and cloud computing workload while achieving real-time monitoring of dairy cow rumination behavior, a real-time monitoring method was proposed for cow ruminant behavior based on edge computing. [Methods] Autonomously designed edge devices were utilized to collect and process six-axis acceleration signals from cows in real-time. Based on these six-axis data, two distinct strategies, federated edge intelligence and split edge intelligence, were investigated for the real-time recognition of cow ruminant behavior. Focused on the real-time recognition method for cow ruminant behavior leveraging federated edge intelligence, the CA-MobileNet v3 network was proposed by enhancing the MobileNet v3 network with a collaborative attention mechanism. Additionally, a federated edge intelligence model was designed utilizing the CA-MobileNet v3 network and the FedAvg federated aggregation algorithm. In the study on split edge intelligence, a split edge intelligence model named MobileNet-LSTM was designed by integrating the MobileNet v3 network with a fusion collaborative attention mechanism and the Bi-LSTM network. [Results and Discussions] Through comparative experiments with MobileNet v3 and MobileNet-LSTM, the federated edge intelligence model based on CA-MobileNet v3 achieved an average Precision rate, Recall rate, F_1 -Score, Specificity, and Accuracy of 97.1%, 97.9%, 97.5%, 98.3%, and 98.2%, respectively, yielding the best recognition performance. [Conclusions] It is provided a real-time and effective method for monitoring cow ruminant behavior, and the proposed federated edge intelligence model can be applied in practical settings.

Key words: cow rumination behavior; real-time monitoring; edge computing; improved MobileNet v3; edge intelligence model; Bi-LSTM

CLC number: S823; TP391.4

Documents code: A

Article ID: SA202405023

Citation: ZHANG Yu, LI Xiangting, SUN Yalin, XUE Aidi, ZHANG Yi, JIANG Hailong, SHEN Weizheng. Real-Time Monitoring Method for Cow Rumination Behavior Based on Edge Computing and Improved MobileNet v3[J]. Smart Agriculture, 2024, 6(4): 29-41. DOI: 10.12133/j.smartag.SA202405023 (in English with Chinese abstract)

张宇, 李相廷, 孙雅琳, 薛爱迪, 张翼, 姜海龙, 沈维政. 基于边缘计算和改进 MobileNet v3 的奶牛反刍行为实时监测方法[J]. 智慧农业(中英文), 2024, 6(4): 29-41. DOI: 10.12133/j.smartag.SA202405023

0 Introduction

The timing and intensity of rumination activities in cows are crucial metrics for assessing their daily behavioral patterns^[1]. Continuous and real-time monitor-

ing of rumination activities is beneficial for maximizing animal welfare and farm productivity^[2]. Currently, the primary methods for monitoring rumination activities in cow are contactless, utilizing machine vision

Received date: 2024-05-30

Foundation items: The National Key Research and Development Program of China (2023YFD2000700); Supported by The Earmarked Fund for CARS36 (CARS36)

Biography: ZHANG Yu, research interests are agricultural AI and AIoT. E-mail: zhangyu1900@neau.edu.cn

*Corresponding author: SHEN Weizheng, PhD, Professor, research interest is smart livestock. E-mail: wzshen@neau.edu.cn

copyright©2024 by the authors

and wireless sensor technology^[3]. Wearable sensor-based monitoring systems have gained increasing popularity due to their cost-effectiveness and ease of integration with wireless networks. The most commonly used sensors include sound sensors, pressure sensors, and velocity sensors. Sound monitoring technology identifies cow rumination activity by analyzing the sounds produced during the rumination process. However, sound recognition has a restricted detection range and is vulnerable to interference from noisy environments, which can compromise system efficacy. The use of hydraulic tubes in pressure sensors to collect cow information can affect their comfort and is susceptible to damage, potentially risking leakage of fluids. Velocity sensor can overcome the limitations of sound sensors and pressure sensors. Tani et al.^[4] utilized a monitoring system equipped with a single-axis acceleration sensor. This system extracts feature patterns from feeding and rumination to distinguish jaw movements and then matches similar feature patterns in unanalyzed activities. However, the accuracy of recording chewing signals is affected by the attachment position of the sensor. Vázquez Diosdado et al.^[5] developed a decision tree algorithm that utilizes three-axis acceleration data collected from sensors mounted on the cow's neck to distinguish behaviors of lying, standing, and eating. Benaissa et al.^[6] also fixed three-axis acceleration sensors on the cow's neck to gather data and devised a simple decision tree algorithm to identify eating and rumination behaviors. Shen et al.^[7] conducted further research on identifying cow rumination behavior using data obtained from three-axis acceleration sensors. Hou^[8] proposed a deep learning model based on cow activity data to recognize cow rumination behavior, building on machine learning techniques. In all the studies mentioned above, the raw data collected by sensors need to be transmitted to a backend system for processing, making it challenging to achieve real-time monitoring of cow rumination behavior. Additionally, the transmission of a large volume of data results in higher energy consumption and shorter battery life for the sensors.

The progression of edge computing is accelerating the trend of shifting from cloud computing to the edge. Edge computing has become a solution that moves cloud services closer to the network edge, closer to data sources, and Internet of Things (IoT) devices^[9-11]. Edge intelligence, utilizing both edge comput-

ing and artificial intelligence technologies, enables the deployment of artificial intelligence (AI) algorithms at the network's edge, that analysis and aggregation occur near the data capture points^[12]. Edge intelligence primarily consists of federated edge and split learning models. The federated edge intelligence model is achieved by deploying federated learning in wireless edge networks^[13]. This model consists of multiple terminal devices and edge servers, collaboratively training an AI model across multiple nodes. Models trained locally on terminal devices are aggregated at the edge server, enabling terminal devices and edge servers to share the AI model^[14]. Split edge intelligence is based on split learning techniques, dividing deep learning models into sub-models and performing distributed training at the edge. It splits the AI model into two parts, with one part deployed on terminal devices near the data input layer, and the other part trained on edge servers^[15].

With the increasing popularity and development of IoT devices in the context of smart farming, cattle farms are now producing a wealth of data. The capability to process data in real-time at the terminal nodes is crucial for these farms. However, terminal devices have limited battery life and computational capabilities, making it a challenge to independently manage tasks that are both energy- and computation-intensive^[16]. Edge computing is an emerging computing model that allows for computation to be executed at the network edge. It supports compute-intensive real-time monitoring applications in resource-constrained cattle farm environments. This approach significantly alleviates the load on network bandwidth and cloud data centers, reducing latency and energy consumption in computation^[17, 18]. Devices upload data to edge servers physically located nearby, offloading compute-intensive and energy-intensive tasks to edge servers. This effectively reduces energy consumption on terminal devices and enables real-time processing of tasks^[19]. Bu and Wang^[20] introduced a smart agricultural land system based on deep reinforcement learning, incorporating the concept of edge computing. This system comprises agricultural data acquisition, edge computing, agricultural data transmission, and cloud computing layers. Shen et al.^[21] fused three-axis accelerometer into edge computing equipment to collect cow rumination information.

To enable the computational tasks to be complet-

ed in real-time at the edge, near the cow data source, thereby reducing data transmission volume and network latency, leveraging the advantages of wearable devices and edge computing, wearable six-axis sensor devices was employed in this research to collect information on cow rumination activities and deep learning algorithm was combined to realize the recognition of rumination behavior.

1 Materials and methods

1.1 Experimental data collection

The experiment was conducted at the Acheng Experimental base of Northeast Agricultural University from May 20 to June 20, 2022. The experimental subjects were 10 healthy Holstein cows in the non-lactating period. Six-axis sensors were used as the terminal devices for six-axis acceleration data collection, with a sampling frequency of 5 Hz. The total duration of the dataset is over 180 hours. The terminal devices were installed on the collars worn by the cows as shown in Fig. 1. The cows were separated in two separate barns, with each cow occupying a fenced enclosure made of iron bars. The experiment involved feeding the cows with a ratio of 3 : 7 of concentrate feed and ryegrass hay twice a day, and providing them with an adequate water supply. Each cowshed was equipped with an infrared night vision camera installed 1.5 m in front and 1.7 m above the ground, totaling 10 cameras to serve as a verification system. These infrared night vision cameras were synchronized with their respective terminal devices in terms of clock time, enabling the continuous recording of cow activities throughout the experiment, which were then stored in the cloud. Through monitoring video footage, it was observed manually that cows exhibited continuous chewing and swallowing actions while at rest, which was considered as the occurrence of rumination behavior.

The perception module of the terminal device consists of the MPU6050, which is a 6-axis sensor that integrates a 3-axis MEMS gyroscope and a 3-axis MEMS accelerometer. It can be connected to third-party digital sensors through the I2C interface and communicate with all registers of the device. The wireless transmission module also utilizes Narrow Band-Internet of Things (NB-IoT) technology to reduce power consumption of the terminal device by transmitting data through network aggregation. The edge server is located at the edge layer and is responsible for handling



Fig. 1 Cow rumination behavior monitoring experimental field and sensor wearing position

service requests through the rational deployment and allocation of computing and storage capabilities at the network edge. The Nvidia Jetson AGX Xavier is chosen as the edge server.

Tensorflow 2.3 was chosen as the deep learning framework for the edge server, and Python 3.6 was employed as the development language. The edge server underwent a firmware flashing process using JetPack 4.2 initially. Subsequently, Miniforge was installed, and environment variables were configured. Finally, the deep learning framework Tensorflow 2.3 was installed.

1.2 Data processing

1.2.1 Pose analysis and calculation

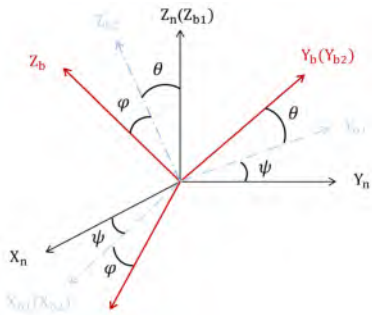
By obtaining the cow's posture information at the current moment, a better understanding of the cow's rumination behavior can be achieved. Pose analysis and estimation were performed using three-axis acceleration and three-axis angular velocity data.

1) Selection of the posture coordinate system and definition of the posture angles. The calculation of the posture angles for cows required a coordinate system transformation, commonly using the body-fixed coordinate system (b-frame) and the navigation coordinate system (n-frame). In this study, local Cartesian coordinates coordinate system was chosen as the navigation coordinate system. The data collection from the terminal device was performed in the body-fixed coordinate system, while the posture calculation was done in the navigation coordinate system. Therefore, a coordinate system transformation was required. The coordinate transformation diagram is shown in Fig. 2. Since the body-fixed coordinate system is fixed to the cow, it changes with the cow's neck posture, thus, the cow's posture changes can be expressed by the coordinate

transformation matrix from the navigation coordinate system to the body-fixed coordinate system, as shown in Equation (1)^[22].

$$\mathbf{T}^b = \mathbf{C}_n^b \mathbf{T}^n \quad (1)$$

Where \mathbf{C}_n^b is the coordinate transformation matrix from the navigation coordinate system to the body-fixed coordinate system; \mathbf{T}^n is the initial state of the body in the navigation coordinate system represented as the attitude vector; \mathbf{T}^b is the attitude vector of the body after the posture change in the body-fixed coordinate system.



Note: X_n pointed to the east; Y_n pointed to the north; Z_n was vertical, pointing upward with respect to the horizontal plane. The body-fixed coordinate system was attached to the cow's neck, with X_b pointing to the right of the body; Y_b pointing forward; Z_b pointing upward, perpendicular to the plane formed by X_b and Y_b . There are three posture angles, namely pitch angle (θ), roll angle (ϕ), and yaw angle (ψ), which represent the orientation of the cow relative to the ground in the cow coordinate system.

Fig. 2 Coordinate schematic diagram of dairy cow posture transitions

In this study, the update of cow posture transformation matrices was accomplished using quaternion methods. This choice was made because quaternion methods could offer advantages over Euler angle methods when dealing with the rotation of the cow's posture, particularly in pitch angle θ induced a deviation of $\pm \frac{\pi}{2}$. Unlike Euler angles, quaternion methods do not encounter issues with singularities and gimbal lock, and they provide a comprehensive representation of posture information in all directions. Additionally, quaternion posture calculations involve lower computational complexity and enable real-time updates of posture angles.

2) Fusion attitude estimation based on the Kalman filtering. Using a single sensor, either an accelerometer or a gyroscope to estimate attitude angles can lead to errors^[23]. Therefore, a Kalman filter algorithm was employed for data fusion to achieve attitude esti-

mation and mutual complementation of multiple sensors. This approach aimed to address noise interference and obtain the optimal estimation of attitude angles.

The corresponding attitude angle information was calculated using the three-axis acceleration measured by the accelerometer, and the observation update equation of the system was established. The angular velocity update equation was then established using the information from the three-axis angular velocity and serves as the state update equation in the Kalman filtering (KF) process. The state update equation is given in Equation (2).

$$\begin{bmatrix} \theta_k \\ \Delta\omega_k \end{bmatrix} = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{k-1} \\ \Delta\omega_{k-1} \end{bmatrix} + \begin{bmatrix} \omega_k dt \\ 0 \end{bmatrix} + \begin{bmatrix} \omega_\theta \\ \omega_\omega \end{bmatrix} \quad (2)$$

Where θ_k is the optimal estimation of attitude angle at time k ; θ_{k-1} is the optimal estimation of attitude angle at the previous time step; ω_k is the measurement value of the gyroscope, $\Delta\omega_k$ is the priori estimate of gyroscope error at time k ; $\Delta\omega_{k-1}$ is the optimal estimation of gyroscope output error; ω_θ and ω_ω are the noises of the attitude system. The observation update equation of the system is established using the corresponding attitude angle information calculated from the three-axis acceleration measured by the accelerometer, and it is given in Equation (3).

$$\theta'_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_k \\ \Delta\omega_k \end{bmatrix} + v_\theta \quad (3)$$

Where θ'_k is the attitude angle information at time k ; v_θ is the measurement noise of the attitude system.

The structure diagram of the attitude angle estimation based on the Kalman filter algorithm is shown in Fig. 3. By iterative computation, the data fusion of the attitude detection system is performed, and the optimal estimation of the attitude angles is ultimately obtained. By collecting three-axis acceleration data, the three-axis angular velocity data was calibrated and compensated to calculate the optimal pitch angle θ and roll angle ϕ .

Fig. 4 shows the pitch angle θ and roll angle ϕ of a cow during rumination, comparing the experimental results obtained with and without the using of Kalman filtering for attitude estimation. Through comparative analysis, it can be observed that Kalman filtering reduces the fluctuations in the estimated attitude angles. The fusion of multisensor data based on Kalman filtering yields better results for attitude estimation compared to using a single sensor alone.

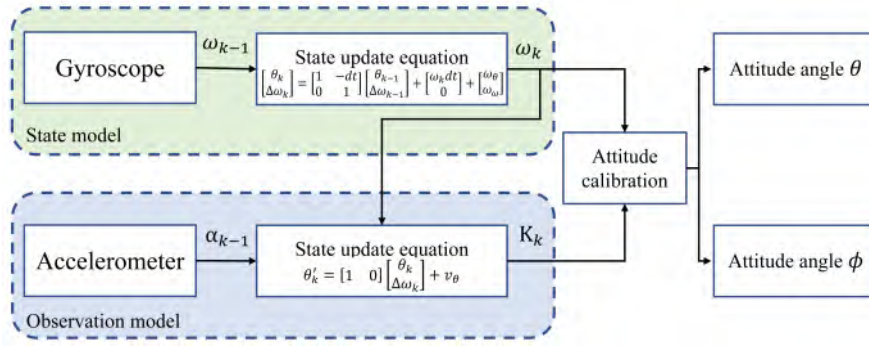


Fig. 3 Block diagram of attitude Angle solution using the Kalman filter algorithm

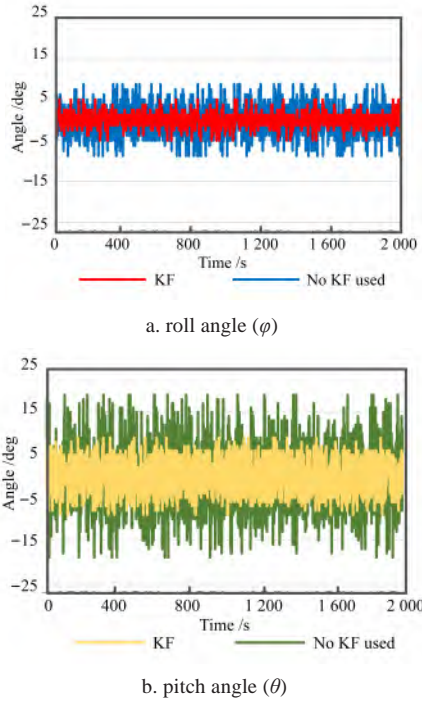


Fig. 4 Comparison diagram of attitude estimation experiments with and without the Kalman filter

1.2.2 Feature extraction and selection

To comprehensively explore the statistical characteristics of the data, both time-domain and frequency-domain features were extracted from the six-axis data. SMV(Support Vector Machine) and attitude angles underwent Fourier transformation to obtain frequency spectrum energy, frequency domain entropy, and the DC component as frequency-domain features. Frequency spectrum energy and frequency domain entropy reflected the energy consumption of rumination behaviors^[24]. The DC component represents the magnitude of the first component obtained after performing a fast Fourier transform. It is related to the reverse gravity acceleration and the components in the x , y , and z axes, thus reflecting the attitude of the cow's

neck.

For the six-axis data, a data feature set consisting of 96 dimensions was extracted, which included both time-domain and frequency-domain features. The feature information is presented in Table 1. The six-axis data was segmented, and a continuous and non-overlapping sequence of 288 frames was selected as the minimum processing unit. Feature extraction was performed within groups using a sliding window of length 16 frames. The sliding window has a window length of 3, resulting in a dataset of 96 frames. After feature extraction, the data set has an input dimension of $96 \times 96 \times 1$.

1.3 Federated edge intelligence model

1.3.1 Improved MobileNet v3 with Co-attention mechanism

The deployment of deep neural network models typically requires high-performance computing hardware support. However, edge devices have limited network resources compared to cloud servers, necessitating the selection of lightweight neural networks as the base network for research on cow rumination behavior recognition methods. In 2017, the Google team proposed the MobileNet network, a network specifically designed for lightweight convolutional neural networks, aimed at enabling neural networks to be deployed on edge devices such as mobile devices and embedded systems. MobileNet constructs lightweight neural networks through depthwise separable convolutions, significantly reducing model parameters and computational requirements with a minor decrease in accuracy compared to traditional convolutional neural networks. MobileNet has evolved into three versions: v1, v2, and v3, becoming an essential tool for neural network applications on mobile devices.

MobileNet v3 incorporates the Squeeze and Excite

Table 1 Time_domain and frequency_domain feature extraction information for six_axis data

Feature types	Feature quantity	Feature description
Minimum value	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
The first quartile	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Median value	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Third quartile	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Maximum value	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Mean value	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Root mean square	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Standard deviation	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Mean absolute deviation	9	Triaxial acceleration, triaxial angular velocity, SMV, attitude Angle
Coefficient of correlation	6	Triaxial acceleration, triaxial angular velocity
Spectral energy	3	SMV, attitude Angle
Frequency domain entropy	3	SMV, attitude Angle
Direct current component	3	SMV, attitude Angle

tation SE attention in some bneck blocks to improve model accuracy by increasing the weights of salient feature channels. However, SE attention ignores positional information and only considers internal channel information. In this study, the CA attention was introduced after the depth wise convolution, which followed the inverted residual module within the bneck structure, thereby forming the CA-bneck. As shown in Fig. 5, the CA-bneck structure embeds positional information into channel attention, thereby reduces additional computational costs and further enhances MobileNet v3's focus on key features.

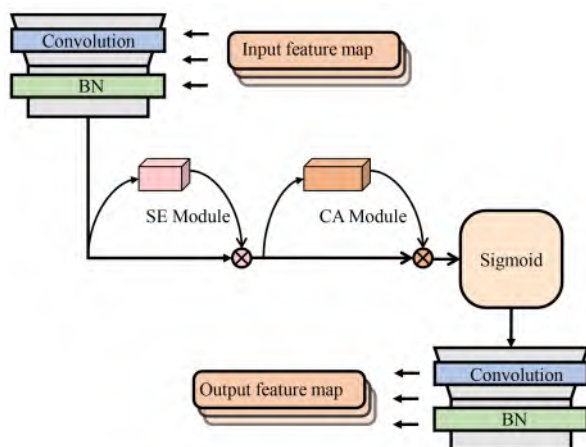


Fig. 5 Structure diagram of CA-bneck

The constructed dataset of size $96 \times 96 \times 1$ was used as the input for the CA-MobileNet v3 network. The structure of the improved fusion cooperative attention mechanism in the CA-MobileNet v3 network is shown in Table 2. The CA-MobileNet v3 network performed downsampling through convolutional stride operations without pooling operations.

Table 2 Structure of CA-MobileNet v3

Input	Operation	Exp size	Output	CA	NL	Step length
$96 \times 96 \times 1$	conv2d, 3×3	—	16	—	HS	2
$48 \times 48 \times 16$	bneck, 3×3	16	16	✓	RE	2
$24 \times 24 \times 16$	bneck, 3×3	72	24	—	RE	2
$12 \times 12 \times 24$	bneck, 3×3	88	24	—	RE	1
$12 \times 12 \times 24$	bneck, 3×3	96	40	✓	HS	2
$6 \times 6 \times 40$	bneck, 5×5	240	40	✓	HS	1
$6 \times 6 \times 40$	bneck, 5×5	240	40	✓	HS	1
$6 \times 6 \times 40$	bneck, 5×5	120	48	✓	HS	1
$6 \times 6 \times 48$	bneck, 5×5	144	48	✓	HS	1
$6 \times 6 \times 48$	bneck, 5×5	288	96	✓	HS	2
$3 \times 3 \times 96$	bneck, 5×5	576	96	✓	HS	1
$3 \times 3 \times 96$	bneck, 5×5	576	96	✓	HS	1
$3 \times 3 \times 96$	conv2d, 1×1	—	576	✓	HS	1
$3 \times 3 \times 576$	pool, 3×3	—	—	—	—	1
$1 \times 1 \times 576$	conv2d 1×1 , NBN	—	1 024	—	HS	1
$1 \times 1 \times 1 024$	conv2d 1×1 , NBN	—	1	—	—	1

Note: Exp size represents the number of 1×1 expansion convolutional kernels in the inverted residual structure; CA indicates whether the improved CA-bneck is used; NL represents the type of activation function; HS denotes the usage of h-swish; RE represents ReLU; NBN indicates that no batch normalization operation is performed.

1.3.2 Federated edge intelligence model based on CA-MobileNet v3

The construction of the federated edge intelligence model involved two main entities: terminal devices and edge servers. Terminal devices collaborate to train and share a lightweight neural model, while the edge server was responsible for collecting local model parameters sent by the terminal devices and performing aggregation. In this study, the terminal de-

viced and edge server shared the fused cooperative attention mechanism of CA-MobileNet v3 and perform collaborative training using distributed cow data samples available on the terminal devices.

The federated edge intelligence system in this study consisted of M terminal devices and a base station equipped with an edge server. The base station was connected to the terminal devices through a wireless channel. The training process of the federated edge intelligence model based on CA-MobileNet v3 follows the flowchart shown in Fig. 6. The learning process of the federated edge intelligence model typically involved four steps:

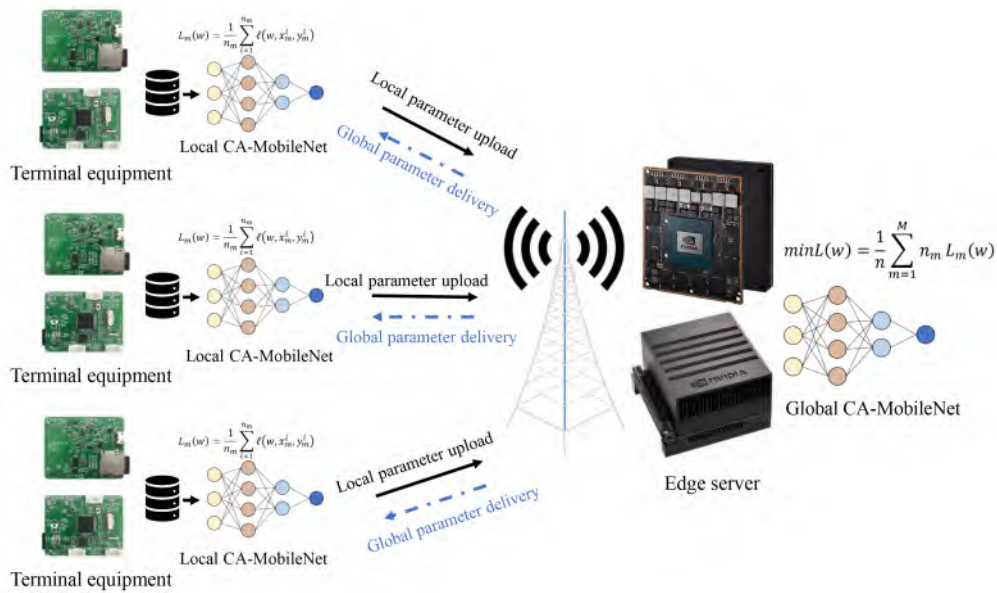


Fig. 6 Flow chart of training of federated edge intelligent model

Through interactions with cattle ranch users, terminal devices obtain labeled training samples. These samples are used as the local dataset as Equation (4).

$$Q_m = \{ (x_m^1, y_m^1), (x_m^2, y_m^2), \dots, (x_m^{n_m}, y_m^{n_m}) \} \quad (4)$$

where x_m^i represents the features of the i -th training sample of the m -th terminal device, y_m^i represents the corresponding sample label indicating whether it is rumination or not, and n_m denotes the number of training samples owned by the m -th terminal device. The training objective of federated learning is to minimize the global loss function $L(w)$. When the CA-MobileNet v3 with the fused cooperative attention mechanism is deployed on the terminal devices, the local loss function of the m -th terminal device on its local dataset Q_m is defined as shown in Equation (5), and the global loss function is defined as shown in Equa-

tion (6).

1) The edge server initialized the relevant parameters, and the terminal devices download the global model as their initial local models;

2) The terminal devices train their local models using the collected real-time local data and upload the model parameters;

3) The edge server collected the local model parameters from each terminal device and performs aggregation to update the global model;

4) The above three steps were repeated until the global model converges, and the edge server deploys the updated model parameters to each terminal device.

tion (6).

$$L_m(w) = \frac{1}{n_m} \sum_{i=1}^{n_m} l(w, x_m^i, y_m^i), \forall m \quad (5)$$

$$L(w) = \frac{1}{n} \sum_{m=1}^M n_m L_m(w) \quad (6)$$

In federated edge intelligence systems, edge servers collect local model parameters from terminal devices and to refine the global model through aggregation. In order to improve the performance of federated learning, the strategy of averaging the parameters was adopted, as implemented in the FedAvg algorithm proposed by Mc Mahan et al^[25], as shown in Equation (7).

$$w_t = \sum_{m=1}^M \frac{|Q_m|}{|Q|} w_t^m \quad (7)$$

Where $|Q_m|$ represents the data size of the local dataset on terminal device m ; $|Q|$ represents the total data size; w_t^m represents the local model parameters of

terminal device m in the r -th communication round; w_t represents the global model parameters at this time.

1.4 Split edge intelligence model

The convolutional recurrent neural network (CRNN) combines the strengths of convolutional neural networks (CNN) and bidirectional long short-term memory (Bi-LSTM) networks to handle tasks that involve sequential data^[26,27]. The MobileNet-RNN model based on the idea of CRNN was proposed. The model consists of three parts: a convolutional neural network module, a recurrent neural network module, and a fully connected module. The CNN module was composed of a lightweight CNN, CA-MobileNet v3, which incorporates a fusion collaborative attention mechanism. Firstly, CA-MobileNet v3 was used to extract features from the six-axis data of cows, reducing computational complexity. Secondly, the extracted features were fed into a Bi-LSTM layer, followed by a fully connected module to recognize the rumination behavior of cows. The fully connected layer consisted of one fully connected layer and one Softmax classification layer. To avoid overfitting, a batch normalization (BN) layer was introduced after the Bi-LSTM output and the fully connected layer. Finally, the Softmax classification layer was used to recognize whether cows were engaged in rumination behavior. The com-

putational process is illustrated in Equations (8) and (9).

$$\hat{R} = \frac{(r_i)^T - \mu}{\sqrt{\sigma^2 + \varepsilon}} \quad (8)$$

$$O = \sigma(W_0 \hat{R} + b_0) \quad (9)$$

Where $R = \{(r_1)^T, (r_2)^T, \dots, (r_i)^T, \dots, (r_M)^T\}$ represents the output results of the Bi-LSTM; \hat{R} denotes the results after batch normalization processing.

To implement split edge intelligence, the MobileNet-LSTM model was proposed, and the overall network structure is depicted in Fig. 7. The MobileNet-LSTM network comprised a CNN module, a RNN module, and a Fully Connected module. The CNN module is deployed on the edge server. CA-MobileNet v3 extracts features from the input dataset, ultimately obtaining a sequence of cow behavior features through feature dimension reduction. This sequence of behavior features was then fed into the RNN module, which consists of a Bi-LSTM network. The Bi-LSTM network captured temporal correlations in cow behavior data. The fully connected module included two fully connected layers and a Softmax classification layer. BN was introduced after each fully connected layer to prevent overfitting. Finally, the softmax classification layer was employed to recognize cow rumination behavior.

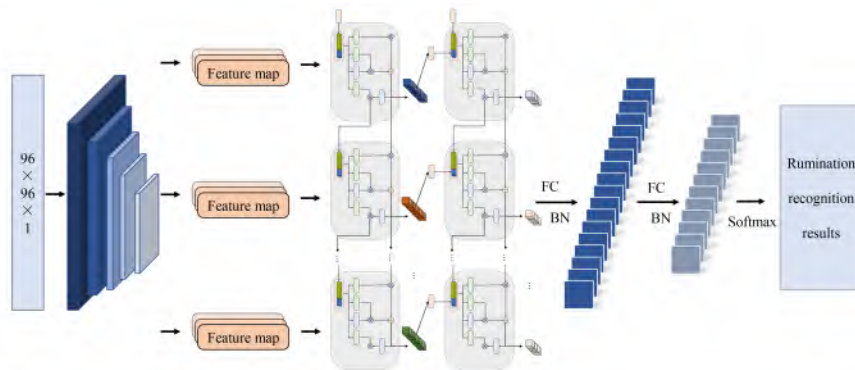


Fig. 7 Structure diagram of MobileNet-LSTM network

To build a distributed edge intelligence model, the MobileNet-LSTM was divided into two parts using split learning techniques. A splitting layer was introduced between the convolutional neural network module and the recurrent neural network module. The shallow network and deep network were trained separately. The shallow network, which consists of the convolutional neural network module, was deployed on the terminal devices for feature extraction. It extracts

useful features from the raw input data. The deep network, including the recurrent neural network module and the fully connected module, was deployed on the edge server. The deep network performed fusion learning among features to extract more complex higher-order features. As shown in Fig. 8, the training process of the split-based edge intelligence model based on MobileNet-LSTM typically involves five steps.

1) Initialization: Each terminal device and the

edge server initialize their respective network models;

2) Data collection and forward propagation: Terminal devices collect data and perform forward propagation until the splitting layer, obtaining the output features of the splitting layer. These features were then uploaded to the edge server;

3) Forward and backward propagation at the edge server: The edge server receives the feature data, performs forward and backward propagation, and obtains

the gradients of the splitting layer. It sent these gradients to all terminal devices;

4) Backward propagation at terminal devices: Terminal devices used the gradients of the splitting layer to perform backward propagation;

5) Iteration: Terminal devices and the edge server iteratively execute the above steps until the model converges.

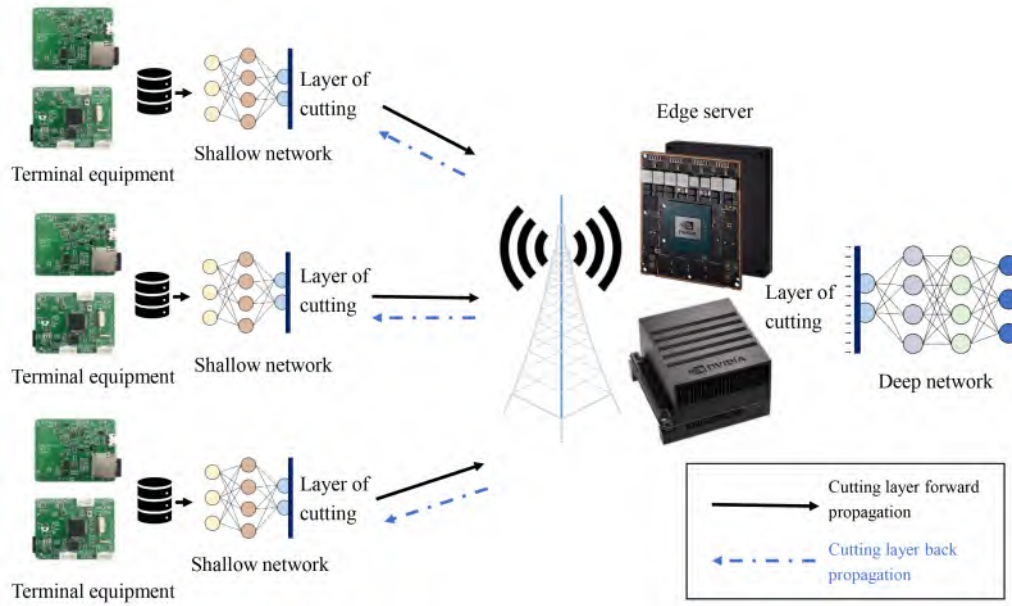


Fig. 8 Training flow chart of split edge intelligence model

1.5 Performance evaluation metrics

The performance evaluation metrics calculated based on the recognition results were effective for measuring the uncertainty between the predicted class and the true class, aiming to evaluate the classifier's performance. In real-time cow rumination behavior recognition tasks, a test data input was assigned to one and only one predefined class, allowing for clear definitions of true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Therefore, performance can be evaluated using metrics such as Precision, Recall, F_1 -Score, Specificity, and Accuracy. The formulas are shown in Equations (10)~(14).

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F_1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

$$Specificity = \frac{TN}{TN + FP} \quad (13)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

2 Results and analysis

2.1 Test results of federated edge intelligence model

In the experiment, a preprocessed dataset that conforms to the input requirements of the neural network model was selected for training. The dataset was divided into training, validation, and testing set in a ratio of 6:2:2. The dataset includes approximately 5 million preprocessed six-axis acceleration data points, with around 3 million data points in the training set and approximately 1 million data points each in the validation and test sets. The CA-MobileNet v3 network was utilized to construct the real-time cow rumination behavior recognition model. During the model training, the cow rumination recognition model was

trained on the edge server using the training set. The validation set was used for parameter adjustment and further training until the model converges. In the model test, the cow rumination recognition model obtained during the training phase based on CA-MobileNet v3 was deployed on both the terminal devices and the edge server. The testing set was then used to evaluate the rumination recognition results.

To validate the role of the collaborative attention module in improving rumination recognition accuracy, a comparison was made between the MobileNet v3 and CA-MobileNet v3 model. The performance evaluation metrics of both models are presented in Table 3. From the data in the table, it can be observed that CA-MobileNet v3 exhibits a relative increase of 5.3% in precision and 2.7% in accuracy compared to MobileNet v3. This suggested that the integration of collaborative attention enhances the model's performance in cow rumination recognition tasks.

Table 3 Performance comparison of MoblieNet v3 and CA-MoblieNet v3 model of federated edge intelligence model

Performance metrics	MoblieNet v3	CA-MoblieNet v3
<i>Precision/%</i>	90.5	95.8
<i>Recall/%</i>	91.6	93.6
<i>F₁-Score/%</i>	91.1	94.7
<i>Specificity/%</i>	94.5	97.7
<i>Accuracy/%</i>	93.5	96.2

The federated edge intelligence model leveraged data to complement each other, effectively increasing the effective data volume. It achieved this without the need to centralize cow data samples, enabling efficient feature extraction and utilization. During the training process of the federated edge intelligence model, the following settings were applied: Training Batch-size 16; number of Epochs 125; Initial Learning Rate 0.001; number of Terminal Devices Participating in Federated Learning System 10.

In the federated learning process, each terminal device uploaded the weight parameters of its local model to the edge server after receiving the new global model. The federated center server was hosted on the edge server, equipped with an NVIDIA GTX 1080Ti GPU, an Intel Core i7-7800X CPU, 64GB of RAM, and 11GB of VRAM.

Comparative experiments were conducted by setting different values for I (the number of iterations for local training on terminal devices) and analyzing the

accuracy and loss curves shown in Fig. 9. From the loss curve, it can be observed that as I increase, the model's convergence speed becomes faster. The accuracy curve revealed that as I increase from 1 to 6, the accuracy of the converged model gradually improves. However, when I increase from 6 to 10, the accuracy of the converged model shows a declining trend. This is because increasing the value of I results in more local iterations of gradient descent training on terminal devices, leading to faster convergence and improved accuracy. However, an excessively large I value leads to too many local training iterations between two model aggregations, causing overfitting to local data and weakening the effectiveness of the FedAvg federated aggregation algorithm.

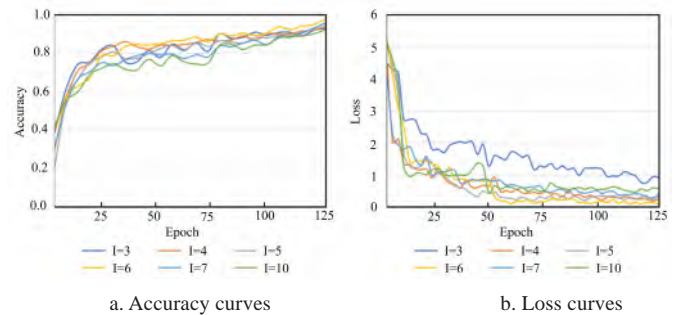


Fig. 9 Accuracy curve and Loss curve of training models with different I values of CA-MobileNet v3

In summary, setting $I = 6$ achieves the best recognition performance, and the corresponding performance evaluation metrics are shown in Table 4. From Table 4, it can be seen that the federated edge intelligence model based on CA-MobileNet v3, after undergoing federated learning, exhibits improvements in performance metrics. Specifically, it shows a 4.6% increase in recall and a 2.4% increase in accuracy. These experimental results indicate that the CA-MobileNet v3 model based on federated learning can effectively enhance the recognition of cow rumination behavior.

Table 4 Performance comparison results of federated learning

Performance metrics	non-federated learning model	federated learning model
<i>Precision/%</i>	95.0	97.1
<i>Recall/%</i>	93.3	97.9
<i>F₁-Score/%</i>	94.1	97.5
<i>Specificity/%</i>	97.2	98.3
<i>Accuracy/%</i>	95.8	98.2

2.2 Test results of split edge intelligence model

The split edge intelligence model was trained in a

supervised manner. To validate the effectiveness of the proposed method, ablation experiments were conducted between CA-MobileNet v3 and MobileNet-LSTM. Both models were configured with the same training parameters: a batch size of 16 125 epochs, and an initial learning rate of 0.001. Ablation experiments on CA-MobileNet v3 and MobileNet-LSTM resulted in loss and accuracy curves for both training and testing, as shown in Fig. 10.

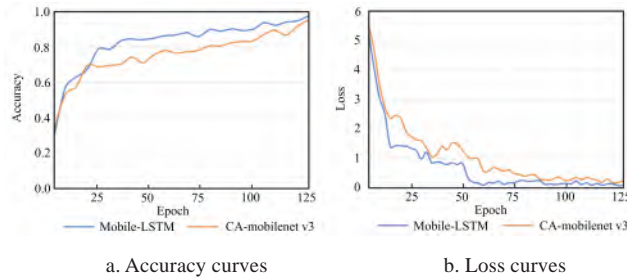


Fig. 10 Comparison of training curves between CA-MobileNet v3 and MobileNet-LSTM of split edge intelligence model

As seen in Fig. 10a, the MobileNet-LSTM, based on the split edge intelligent model, achieves a final recognition accuracy of 96.2%. This was an improvement compared to the 95.8% accuracy achieved by CA-MobileNet v3. The higher accuracy of MobileNet-LSTM was attributed to its fusion of contextual information related to cow behavior, which enhances the recognition accuracy. From Fig. 10b, it can be observed that MobileNet-LSTM converges faster compared to CA-MobileNet v3. This is because MobileNet-LSTM incorporates a BN layer, which speeds up the training of the network.

2.3 Experimental contrastive analysis

The federated edge intelligence model based on CA-MobileNet v3 and the split edge intelligence model based on MobileNet-LSTM were selected for comparative experiments. The performance indicators of the experimental results are shown in Table 5.

The federated edge intelligent model based on

Table 5 Comparison of the performance of federated and split edge intelligence models

Performance metrics/%	Federated edge intelligence model	Split edge intelligence model
<i>Precision</i>	97.1	95.8
<i>Recall</i>	97.9	93.7
<i>F₁-Score</i>	97.5	94.8
<i>Specificity</i>	98.3	97.7
<i>Accuracy</i>	98.2	96.2

CA-MobileNet v3 outperforms the split edge intelligent model based on MobileNet-LSTM, leading to further improvements in performance metrics. This is because the federated learning-based model utilizes local training and uploads model parameters, aggregates the parameters to update the global model until convergence, and can extract deeper cow rumination behavior features. On the other hand, the split edge intelligent model required substantial intermediate data transmission between the terminal devices and the edge layer, which may result in data loss and a decrease in recognition accuracy.

Fig. 11 presents the experimental results in the form of box plots for MobileNet v3, CA-MobileNet v3, the federated edge intelligent model, and the split edge intelligent model. Through experimental comparisons, it can be observed that the federated learning-based CA-MobileNet v3 network, i.e., the federated edge intelligent model, not only improves the recognition accuracy of rumination behavior but also led to a more concentrated data distribution. Compared to the split edge intelligent model based on MobileNet-LSTM, the federated edge intelligent model based on federated learning and CA-MobileNet v3 exhibited a more stable and concentrated prediction distribution, with fewer outliers. This indicates that the model has greater reliability and effectiveness in recognizing cow rumination behavior. Although the method proposed by Shen et al.^[7] achieved a recall of 94.3%, which was better than that of this research, it was at the cost of transmitting large amounts of data and increasing equipment energy consumption. In contrast, although the performance indices of this study have declined slightly, this method have reduced the amount of data transmission and cloud computing to achieve real-time cow rumination recognition in an environment with low network performance.

3 Conclusions

In this study, based on the concept of edge computing, proposed the use of edge devices to capture and process real-time six-axis acceleration signals of cows. By integrating a cooperative attention mechanism into the MobileNet v3 network, the CA-MobileNet v3 network was introduced. The federated edge intelligent model was subsequently constructed utilizing the CA-MobileNet v3 network in conjunction with the FedAvg model aggregation algorithm.

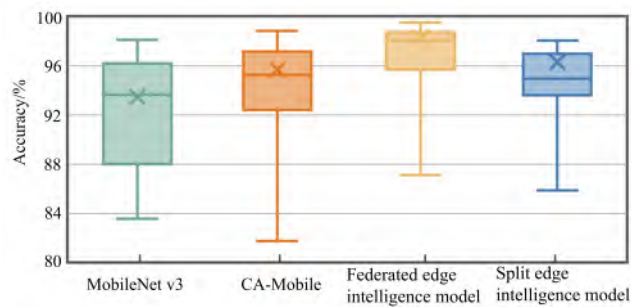


Fig. 11 Boxplot of experimental results distribution for MobileNet v3, CA-MobileNet, Federated edge intelligence model, and Split edge intelligence model

Experimental findings reveal that the proposed CA-MobileNet v3 network enhances precision by 5.3% compared to the MobileNet v3 network, while the FedAvg federated aggregation algorithm boosts the recall rate by 4.3% within the federated edge intelligent model, underscoring the efficacy of the proposed federated edge intelligent model. Furthermore, leveraging the CA-MobileNet v3 network and the Bi-LSTM network, a split edge intelligent model based on MobileNet-LSTM was designed, and comparative experiments were conducted between the federated edge and split edge intelligent models. The experimental results show that the federated edge intelligent model achieves the best recognition performance, with average Precision, Recall, F_1 -Score, Specificity, and Accuracy for cow rumination behavior reaching 97.1%, 97.9%, 97.5%, 98.3%, and 98.2%, respectively.

The proposed edge intelligence model not only effectively expands the dimensionality of cow data samples but also improves the accuracy of cow rumination behavior recognition.

Conflict of interest statement: All authors declare no competing interests.

References:

- [1] BRAUN U, TSCHONER T, HÄSSIG M. Evaluation of eating and rumination behaviour using a noseband pressure sensor in cows during the peripartum period[J]. BMC veterinary research, 2014, 10: ID195.
- [2] HE M N. Research on key technologies for digital management of cattle[D]. Jinan: Shandong University.
- [3] JI N, YIN Y L, SHEN W Z, et al. Pig sound analysis: A measure of welfare[J]. Smart Agriculture, 2022, 4(2): 19-35.
- [4] TANI Y, YOKOTA Y, YAYOTA M, et al. Automatic recognition and classification of cattle chewing activity by an acoustic monitoring method with a single-axis acceleration sensor[J]. Computers and electronics in agriculture, 2013, 92: 54-65.
- [5] VÁZQUEZ DIOSDADO J A, BARKER Z E, HODGES H R, et al. Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system[J]. Animal biotelemetry, 2015, 3(1): 15.
- [6] BENAÏSSA S, TUYTTENS F A M, PLETS D, et al. Classification of ingestive-related cow behaviours using Rumi-Watch halter and neck-mounted accelerometers[J]. Applied animal behaviour science, 2019, 211: 9-16.
- [7] SHEN W Z, CHENG F, ZHANG Y, et al. Automatic recognition of ingestive-related behaviors of dairy cows based on triaxial acceleration[J]. Information processing in agriculture, 2020, 7(3): 427-443.
- [8] HOU S. Research on real-time recognition of cattle ruminating behavior based on activity data and neural networks[D]. Hohhot: Inner Mongolia University, 2021.
- [9] WANG J H, HAN Y X. Cognitive radio sensor networks clustering routing algorithm for crop phenotypic information edge computing collection[J]. Smart agriculture, 2020, 2(2): 28-47.
- [10] CHIANG M, ZHANG T. Fog and IoT: An overview of research opportunities[J]. IEEE Internet of Things journal, 2016, 3(6): 854-864.
- [11] SHI W S, CAO J, ZHANG Q, et al. Edge computing: Vision and challenges[J]. IEEE Internet of Things journal, 2016, 3(5): 637-646.
- [12] ZHU Y. Network public opinion prediction and control based on edge computing and artificial intelligence new paradigm[J]. Wireless communications and mobile computing, 2021, 2021: ID 5566647.
- [13] BONAWITZ K, EICHNER H, GRIESKAMP W, et al. Towards federated learning at scale: system design[J]. Proceedings of the 2019 conference on systems and machine learning, 2019, 1: 374-388.
- [14] WANG T, SUN B, ZHANG Y L, et al. A method for device assessment and federated learning importance aggregation based on edge intelligence, system, device, and readable storage medium[P]. CN112181666A, 2021.
- [15] DUAN Q, HU S, DENG R, et al. Combined federated and split learning in edge computing for ubiquitous intelligence in Internet of Things: State-of-the-art and future directions[J]. Sensors (basel, Switzerland), 2022, 22(16): ID 5983.
- [16] FATEMI MOGHADDAM F, MOGHADDAM S G, ROUZBEH S, et al. A scalable and efficient user authentication scheme for cloud computing environments[C]// 2014 IEEE REGION 10 SYMPOSIUM. Piscataway, New Jersey, USA: IEEE, 2014: 508-513.
- [17] HUANG C, KE Y, HUA X, et al. Current applications and prospects of edge computing in smart agriculture[J]. Transactions of the Chinese society of agricultural engineering, 2022, 38(16): 224-234.
- [18] CHEN G, WANG Z X, WANG J. An AIOT-based comprehensive application system for smart agriculture[J]. Automation applications, 2022(7): 48-50.
- [19] LI Y H. Design and implementation of an optimization system for real-time data processing applications in edge computing mode[M]. Nanjing: Southeast University, 2019.
- [20] BU F Y, WANG X. A smart agriculture IoT system based on deep reinforcement learning[J]. Future generation computer systems, 2019, 99: 500-507.

- [21] SHEN W Z, SUN Y L, ZHANG Y, et al. Automatic recognition method of cow ruminating behaviour based on edge computing[J]. Computers and electronics in agriculture, 2021, 191: ID 106495.
- [22] WEN K, YU K G, LI Y B, et al. A new quaternion Kalman filter based foot-mounted IMU and UWB tightly-coupled method for indoor pedestrian navigation[J]. IEEE transactions on vehicular technology, 2020, 69(4): 4340-4352.
- [23] DISSANAYAKE G, SUKKARIEH S, NEBOT E, et al. The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications[J]. IEEE transactions on robotics and automation, 2001, 17(5): 731-747.
- [24] LI Y G, ZHANG S G, ZHU B, et al. Accurate human activity recognition with multi-task learning[J]. CCF transactions on pervasive computing and interaction, 2020, 2(4): 288-298.
- [25] MC MAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]// International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, Florida, USA: AISTATS, 2017:1273-1282.
- [26] SHI B, BAI X, YAO C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(11): 2298-2304.
- [27] XIONG F, CHEN T, BIAN B C, et al. Chip surface character recognition based on convolutional recurrent neural network[J]. Journal of Zhejiang university, 2023, 57(5): 1-9.

基于边缘计算和改进 MobileNet v3 的奶牛反刍行为实时监测方法

张 宇¹, 李相廷¹, 孙雅琳², 薛爱迪^{1,3}, 张 翼¹, 姜海龙¹, 沈维政^{1*}

(1. 东北农业大学 电气与信息学院, 黑龙江哈尔滨 150030, 中国; 2. 哈尔滨航天恒星数据系统科技有限公司, 黑龙江哈尔滨 150030, 中国; 3. 哈尔滨电机厂有限责任公司, 黑龙江哈尔滨 150030, 中国)

摘 要: [目的/意义] 随着奶牛养殖业向规模化、精准化和信息化养殖迅速发展, 对奶牛健康的监测和管理需求也日益增加。实时监测奶牛的反刍行为对于第一时间获取奶牛健康的相关信息以及预测奶牛疾病具有至关重要的意义。目前, 针对奶牛反刍行为的监测已经提出了多种策略, 包括基于视频监控、声音识别、传感器监测等方法, 但是这些方法普遍存在实时性不足的问题。为了减轻数据传输的数量与云端计算量, 实现对奶牛反刍行为的实时监测, 基于边缘计算的思想提出了一种实时对奶牛反刍行为进行监测的方法。[方法] 使用自主设计的边缘设备实时地采集并处理奶牛的六轴加速度信号, 基于六轴数据提出了基于联邦式与拆分式边缘智能这两种不同的策略对奶牛反刍行为实时识别方法展开研究。在基于联邦式边缘智能的奶牛反刍行为实时识别方法研究中, 通过协同注意力机制改进 MobileNet v3 网络提出了 CA-MobileNet v3 网络, 进而利用 CA-MobileNet v3 网络和 FedAvg 模型聚合算法, 设计了联邦式边缘智能模型。在基于拆分式边缘智能的奶牛反刍行为实时识别方法研究中, 利用融合协同注意力机制的 MobileNet v3 网络和 Bi-LSTM 网络, 设计了基于 MobileNet-LSTM 的拆分式边缘智能模型。[结果和讨论] 对比了 MobileNet v3、CA-MobileNet、联邦式边缘智能模型, 以及拆分式边缘智能模型的识别准确率, 其中基于 CA-MobileNet v3 的联邦式边缘智能模型的平均查准率、召回率、 F_1 -Score、特异性以及准确率分别达到 97.1%、97.9%、97.5%、98.3% 和 98.2%, 达到了最佳识别效果。[结论] 本研究为奶牛反刍行为的监测提供了一种实时有效的方法, 所提出的方法可以在实际应用中使用的。

关键词: 奶牛反刍行为; 实时监测; 边缘计算; 改进 MobileNet v3; 边缘智能模型; Bi-LSTM

基金资助: 国家重点研发计划项目 (2023YFD2000700); 财政部和农业农村部: 国家现代农业产业技术体系资助 (CARS36)

作者简介: 张 宇, E-mail: zhangyu1900@neau.edu.cn

***通信作者:** 沈维政, E-mail: wzshen@neau.edu.cn

(登录 www.smartag.net.cn 免费获取电子版全文)